Úvod do TI 2010

Vzorové riešenia úlohy 10.1

Túto úlohu riešilo dosť málo ľudí. Body sa strhávali zá absenciu prechodovej funkcie alebo dostatočného popisu. Mnohí ste použili viacpáskové stroje, čo nebol zámer úlohy. Ale keďže to vyslovene zakázané nebolo a určite všetci viete vysypať dôkaz ekvivalencie so štandardným modelom, body som nestŕhal. Táto úloha by sa dala jednoduchšie riešiť nedeterministickým strojom, avšak nám ide hlavne o to, aby sme si precvičili deterministické Turingove stroje, a preto je aj naše riešenie deterministické.

Pri návrhu jednopáskového Turingovho stroja na jazyk slov, ktorých dĺžka je štvorec prirodzeného čísla, môžeme zvoliť viacero taktík. Jednou z nich je použiť známy fakt, že pre každé prirodzené k platí $(k+1)^2 - k^2 = 2k+1$. Inými slovami, že rozdiel dvoch po sebe idúcich štvorcov je nepárne číslo, ktoré sa zakaždým zväčšuje o 2. To znamená, že ak zoberieme najmenší štvorec 0, a budeme k nemu postupne pridávať 1, potom 3, potom 5 a tak ďalej, budeme dostávať práve 0,1,4,9... Stroj by mohol fungovať nasledovne: postupne by jednotky na vstupe zľava prepisoval na nejaký špeciálny znak. Najskôr by prepísal jednu najľavejšiu, potom ďalšie 3, potom ďalších 5, a tak ďalej. Ak je počet jednotiek na vstupe štvorec prirodzeného čísla, potom sa jednotky minú akurát na konci fázy pripisovania nejakého počtu jednotiek. V opačnom prípade sa minú niekedy uprostred.

Náš návrh bude trošku zložitejší na implementáciu, avšak úplne priamočiary. Na pásku za vstup napíšeme oddeľovaciu #. Za mriežku si budeme písať jednotky, ktorých počet bude kandidát na odmocninu dĺžky vstupu. Na začiatku bude jedna, neskôr pripíšeme ďalšiu a takto budeme počet stále zväčšovať. Technický detail: na ľavom kraji slova budeme mať zapísaný ¢ (ten tam máme z definície) a na pravom konci budeme mať \$.

Keď máme na pravej strane zapísaný nejaký počet jednotiek, teda na páske je slovo tvaru ¢$1\dots1\#1\dots1$\$, kde pred mriežkou je počet jednotiek x a za ňou y, potom začne fáza overovania, či náhodou neplatí $x = y^2$. Keď toto overovanie dopadne úspešne, stroj akceptuje. Ak počas overovania zistíme, že $x > y^2$, potom vidíme, že musíme skúsiť vyššie číslo, a preto pokračujeme zvýšením počtu jednotiek za mriežkou a novým overovaním. Ak nastane tretia možnosť $x < y^2$, potom vieme, že žiadne vyššie číslo y nemá zmysel skúšať a keďže pre každé menšie dopadlo overovanie neúspešne, môžeme prehlásiť, že slovo na vstupe nemá dĺžku štvorca prirodzeného čísla.

A ako presne vyzerá overovanie? Budeme používať špeciálne označené jednotky. Teda budeme mať páskové symboly 1, a okrem toho $\underline{1}$, $\overline{1}$ a $\underline{\overline{1}}$. Majme teda slovo ¢$1\dots1\#1\dots1$\$, kde jednotiek naľavo od # je x a tých napravo je y. Budeme postupne podčiarkovať jednotky naľavo, pričom ich musíme označiť presne $y \cdot y$. Preto spravíme y fáz, počas každej označíme práve y jednotiek. Na začiatku jednej fázy teda máme na pravej strane samé nepodčiarknuté jednotky. Podčiarkneme jednu napravo, potom prejdeme do časti na ľavo od mriežky a tam tiež podčiarkneme jednu jednotku. Potom sa vrátime a zopakujeme to s ďalšou jednotkou. Fáza skončí, ak sme na pravej strane označili všetky jednotky. Počet fáz si sledujeme pomocou nadčiarknutých jednotiek. Nadčiarkovanie a podčiarkovanie sú dve nezávislé spôsoby označovania a pamätania si. Keď skončí fáza, nadčiarkeme si ďalšiu jednotku na pravej strane a zároveň všetkým zrušíme podčiarknutie. Potom pokračujeme ďalšou fázou. Teda, ak máme naľavo 9 jednotiek a napravo 3, potom je na páske na začiatku overovania:

$$¢111111111\#111\$$$

Následne spustíme prvú fázu, ktorá prekopíruje tri jednotky, na jej konci bude na páske:

$$¢\underline{111}111111\#\underline{111}\$$$

Potom si musíme nadčiarknutím zaznačiť, že sme jednu fázu skončili a zároveň odškrtnúť jednotky napravo. Na konci tohto procesu bude páska vyzerať takto:

$$\not{c}\underline{111}111111\#11\overline{1}\$$$

Následne vykonáme druhú fázu, na jej konci bude páska vyzerať takto:

$$\not{c}\underline{111111}111\#\underline{11\overline{1}}\$$$

Následne si zaznačíme, že skončila druhá fáza a znovu vyčistíme pravú stranu:

$$\not{c}\underline{111111}111\#1\overline{11}\$$$

A takto postupujeme, až kým vo všeobecnosti nenadčiarkeme poslednú jednotku napravo - vtedy vieme, že sme vykonali práve y fáz, a teda počet označených jednotiek na ľavej strane je y^2. Teraz už len skontrolujeme, či je naľavo nezaškrtnutá jednotka. Ak by bola, potom zopakujeme overovanie s o jedna väčším číslom.

Formálne, Turingov stroj A je teda sedmica $(Q, \{1\}, \{1, \overline{1}, \underline{1}, \underline{\overline{1}}, \$, \sqcup, \#, \not{c}\}, \delta, q_{zac}, q_{accept}, q_{reject})$. Množina stavov $Q = \{q_{zac}, q_{init1}, q_{init2}, q_{init3}, q_{prenes1},$ $q_{prenes2}, q_{pisem}, q_{vraciam}, q_{novafaza1}, q_{novafaza2}, q_{novafaza3}, q_{kontrola}, q_{pridaj1},$ $q_{pridaj2}, q_{accept}, q_{reject}\}$. Funkciu δ rozpíšeme podľa fáz, ktoré vykonávaju jednotlivé skupiny pravidiel. Na úvod máme nasledovnú sadu pravidiel:

$$\begin{aligned}
\delta(q_{zac}, \not{c}) &= (q_{zac}, \not{c}, +1) \\
\delta(q_{zac}, \sqcup) &= (q_{accept}, \sqcup, 0) \\
\delta(q_{zac}, 1) &= (q_{init1}, 1, +1) \\
\delta(q_{init1}, 1) &= (q_{init1}, 1, +1) \\
\delta(q_{init1}, \sqcup) &= (q_{init2}, \#, +1) \\
\delta(q_{init2}, \sqcup) &= (q_{init3}, 1, +1) \\
\delta(q_{init3}, \sqcup) &= (q_{prenes1}, \$, -1)
\end{aligned}$$

Slovný popis týchto pravidiel: v stave q_{zac} si len skontrolujeme, či nemáme na vstupe prázdne slovo. V opačnom prípade dopíšeme pomocou stavov q_{init1}, q_{init2}, q_{init3} za vstup znaky #1\$ a prejdeme hlavou na jednotku, kde sa začína procedúra overovania. Overovanie správnosti počtu jednotiek na pravej strane realizujeme pomocou nasledovných pravidiel:

$$\begin{aligned}
\delta(q_{prenes1}, 1) &= (q_{prenes2}, \underline{1}, -1) \\
\delta(q_{prenes2}, X) &= (q_{prenes2}, X, -1), \forall X \in \{1, \#\} \\
\delta(q_{prenes2}, X) &= (q_{pisem}, X, +1), \forall X \in \{\not{c}, \underline{1}\} \\
\delta(q_{pisem}, 1) &= (q_{vraciam}, \underline{1}, +1) \\
\delta(q_{pisem}, \#) &= (q_{reject}, \#, 0) \\
\delta(q_{vraciam}, X) &= (q_{vraciam}, X, +1), \forall X \in \{\#, 1, \overline{1}\} \\
\delta(q_{vraciam}, \underline{1}) &= (q_{prenes1}, \underline{1}, -1) \\
\delta(q_{prenes1}, \#) &= (q_{novafaza1}, \#, +1) \\
\delta(q_{novafaza1}, \underline{1}) &= (q_{novafaza1}, 1, +1) \\
\delta(q_{novafaza1}, \underline{\overline{1}}) &= (q_{novafaza1}, \overline{1}, +1) \\
\delta(q_{novafaza1}, \$) &= (q_{novafaza2}, \$, -1) \\
\delta(q_{novafaza2}, \overline{1}) &= (q_{novafaza2}, \overline{1}, -1) \\
\delta(q_{novafaza2}, 1) &= (q_{novafaza3}, \overline{1}, +1) \\
\delta(q_{novafaza3}, \overline{1}) &= (q_{novafaza3}, \overline{1}, +1) \\
\delta(q_{novafaza3}, \$) &= (q_{prenes1}, \$, -1) \\
\delta(q_{novafaza2}, \#) &= (q_{kontrola}, \#, -1)
\end{aligned}$$

Slovný popis: v stave $q_{prenes1}$ si podčiarkneme jednotku pod hlavou a putujeme pomocou stavu $q_{prenes2}$ na ľavú stranu slova, kde pomocou stavu q_{pisem} podčiarkneme najľavejšou nepodčiarknutú jednotku. Následne sa pomocou stavu $q_{vraciam}$ vrátime na do pravej časti a opakujeme to s ďalšou jednotkou. Ak sme preniesli uz všetky jednotky z pravej časti, potom pomocou stavov $q_{novafaza1}$, $q_{novafaza2}, q_{novafaza3}$ odstránime podčiarknutie jednotiek napravo a nadčiarknutím si zaznačíme, že sme dokončili ďalšiu fázu. Keď dokončíme poslednú fázu, používame nasledovné pravidlá:

$$
\begin{aligned}
\delta(q_{kontrola}, \underline{1}) &= (q_{kontrola}, \underline{1}, -1)\\
\delta(q_{kontrola}, \not{c}) &= (q_{accept}, \not{c}, 0)\\
\delta(q_{kontrola}, 1) &= (q_{pridaj1}, 1, -1)\\
\delta(q_{pridaj1}, X) &= (q_{pridaj1}, X, -1), \forall X \in \{\underline{1}, 1\}\\
\delta(q_{pridaj1}, \not{c}) &= (q_{pridaj2}, \not{c}, +1)\\
\delta(q_{pridaj2}, \#) &= (q_{pridaj2}, \#, +1)\\
\delta(q_{pridaj2}, X) &= (q_{pridaj2}, 1, +1), \forall X \in \{1, \underline{1}, \overline{1}\}\\
\delta(q_{pridaj2}, \$) &= (q_{pridaj2}, 1, +1)\\
\delta(q_{pridaj2}, \sqcup) &= (q_{prenes1}, \$, -1)
\end{aligned}
$$

Slovný popis pravidiel: v stave $q_{kontrola}$ hľadáme na ľavej strane nepodčiarknutú jednotku. Ak takú jednotku nenájdeme, akceptujeme. Ak takú nájdeme, znamená to, že počet jednotiek na pravo od # je príliš malý a treba skúsiť väčší. Vtedy pomocou stavov $q_{pridaj1}, q_{pridaj2}$ vyčistíme pásku, zväčšíme počet jednotiek na pravej strane a spustíme nové overovanie.

Keďže definícia Turingovho stroja nám prikazuje, aby bola δ funkcia úplná, musíme ešte dodefinovať všetky ostatné možnosti. Stačí, ak ale povieme, že pre všetky ostatné možnosti stroj A prejde do stavu q_{reject} bez pohybu hlavy a zmeny páskového symbolu.

Vzorové riešenie úlohy 10.2

Časté chyby

Veľa z Vás má problém v tom, aký je rozdiel medzi TS počítajúcim funkciu f a TS akceptujúcim jazyk. Zatiaľ čo prvý TS zoberie vstup x, začne niečo počítať a na záver keď skončí (v akceptačnom stave) bude na páske $f(x)$ (ak by nemal akceptovať, tak by to znamenalo, že x nepatrí do definičného oboru funkcie f), TS akceptujúci jazyk povie o vstupe, či patrí do daného jazyka, alebo nie. Preto tí, čo ste si interpretovali + ako „zjednotenie vstupov" dostali najviac 6 bodov (lebo robili stroj, ktorý akceptoval jazyk a nie stroj, ktorý počítal funkciu). „+" v zadaní bolo myslené ako sčítanie (ale ak si to niekto vysvetlil ako zreťazenie, uznal som to – ale iba teraz, nabudúce už nie).

Okrem tohto nedostatku ste väčšinou strácali body za nedostatočne podrobný popis riešenia.

Riešenie na jednopáskovom TS

V tomto riešení sme predpokladali, že čísla sme kódovali v binárnej sústave (uznával som aj riešenia ktoré používali aj iné sústavy).

Nech $M_1 = (Q_1, \{0,1\}, \Gamma_1, \delta, q_0, q_{acc}, q_{rej})$ je TS ktorý počíta funkciu f_1 a $M_2 = (Q_2, \{0,1\}, \Gamma_2, \delta, q_0, q_{acc}, q_{rej})$ je TS ktorý počíta funkciu f_2. Obidva upravíme nasledovne:

- Premenujeme stavy Q_1 a Q_2 tak, aby $Q_1 \cap Q_2 = \{q_{acc}, q_{rej}\}$ (inými slovami, stavy oboch automatov sú až na akceptačný a odmietací stav rovnaké). Zároveň si zavedieme nový pracovný symbol # (predpokladáme že nie je v Γ_1 ani v Γ_2), ku ktorému sa oba automaty budú správať ako k symbolu $\not{c}$. Oznažme si počiatočný stav TS M_i ako M_i^{start}.

- Premenujeme akceptačný stav stroja M_i na M_i^{acc} (Dôležité: tieto dva stavy nie sú akceptačnými stavmi pre TS, ktorý ideme vybudovať).

Teraz ideme spraviť turingov stroj M ktorý bude počítať funkciu $f_1(x) + f_2(x)$.

1. Páska je v tvare ¢$x\sqcup$. Na koniec pásky zapíšeme # a skopírujeme za to x.
2. Páska je v tvare ¢$x\#x\sqcup$. Posunieme hlavu automatu na # a sputíme automat M_1 (tak, že sa presunieme do stavu M_1^{start}). Keď sa automat dostane do stavu M_1^{acc}, tak sa posunieme na koniec pásky.
3. Páska je v tvare ¢$x\#f_1(x)\sqcup$. Na koniec pásky zapíšeme # a za tento symbol skopírujeme x.
4. Páska je v tvare ¢$x\#f_1(x)\#x\sqcup$. Posunieme hlavu na druhý symbol # a obdobne ako v kroku 2 spustíme automat M_2. Keď sa automat M_2 dostane do stavu M_2^{acc}.
5. Páska je v tvare ¢$x\#f_1(x)\#f_2(x)\sqcup$. Vymažeme z pásky počiatočný prefix $x\#$ tak, že zvyšok vstupu poposúvame na začiatok pásky.
6. Páska je v tvare ¢$f_1(x)\#f_2(x)$. Potrebujeme to už len sčítať. Na koniec pásky dáme # a za tento symbol budeme písať výsledok $(f_1(x) + f_2(x))$. V stave si budeme pamätať celé čísla od 0 po 3 (budeme simulovať sčítavanie v dvojkovej sústave – sčítame dve cifry a prirátame k tomu zvyšok čo nám ostal z predhcádzajúcej iterácie).
   (a) V stave si zapamätáme číslo 0.
   (b) Nájdeme posledný neoznačený symbol z $f_1(x)$, označíme ho a pripočítame ho k číslu v stave (v stave si pamätáme teraz ich súčet). Ak taký symbol neexistuje, tvárime sa ako keby existoval a bola to 0.
   (c) Nájdeme posledný neoznačený symbol z $f_2(x)$, označíme ho a pripočítame ho k tomu symbolu, čo si v stave pamätáme a výsledok si zapamätáme v stave. Ak taký symbol neexistuje, tvárime sa ako keby existoval a bola to 0.
   (d) Prídeme na koniec pásky a symbol $\sqcup$ prepíšeme na 0, ak si v stave pamätáme párne číslo a 1 ak si v stave pamätáme nepárne číslo. Ak si v stave pamätáme 3 alebo 2, tak si budeme v stave pamätať 1 a ináč 0 (vydelili sme číslo v stave dvoma). Ak existuje nejaký neoznačený symbol 0 alebo 1 na páske pred druhým znakom #, tak pokračujeme krokom 2.
   (e) Na páske si pamätáme: ¢$\underline{f_1(x)}\#\underline{f_2(x)}\#(f_1(x) + f_2(x))^R\sqcup$. Všimnite si, že výsledok sme písali na pásku tak, že je reverznutý. Rovnako ako vyššie, všetko pred oboma znakmi # (vrátane nich) vymažeme tak, že zvyšok pásky posunieme na začiatok. Dostaneme ¢$(f_1(x) + f_2(x))^R\sqcup$ ktoré nám už stačí iba reverznúť a môžeme akceptovať.

Riešenie na viacpáskovom TS

Na prednáške ani na cvičeniach sme nikde nemali definované čo je to viacpáskový TS počítajúci funkciu (ale viacpáskový TS ktorý akceptuje jazyk poriadne definovaný bol). Preto sa od Vás v riešení očakávalo, že budete robiť s obyčajnými TS, nie viacpáskovými. Keďže sa ale vyskytli riešenia, ktoré robili s viacpáskovými strojmi, tak uvádzam aj také.

Keďže pracujeme s viacpáskovými strojmi, budeme predpokladať, že aj M_1 a M_2 sú viacpáskové. Ako funguje viacpáskový TS počítajúci funkciu? Namiesto čisto vstupnej pásky bude mať vstupno-výstupnú pásku. Na túto pásku sa môže písať, ale môžu sa zapisovať len symboly vstupnej abecedy (a ¢ a $\sqcup$). Ak stroj počíta funkciu f, tak po skončení výpočtu bude na vstupnej páske[1] hodnota $f(x)$. Obsahy ostatných pások ignorujeme.

Toto riešenie nie je úplné, ale chýbajúce časti sa dajú doplniť z riešenia pre obyčajný TS.

Máme viacpáskové TS M_1, M_2 počítajúce funkcie f_1 a f_2. Nech stroj i používa k_i pások. Nech $k = \max(k_1, k_2) + 3$. Použijeme k pások. Postup:

[1] Mala by sa vlastne volať vstupno-výstupná páska, ale to budeme ignorovať:-)

1. Skopírujeme vstupnú pásku na pásku $k-2$. Na vstupnej páske sa posunieme na začiatok, a spustíme stroj M_1. Ten je upravený tak ako v predchádzajúcom riešení a navyše vie používať aj dalšie pásky (aby bol k páskový) na ktorých ale nič nemení ani sa neposúva.
2. Na vstupnej páske budeme mať $f_1(x)$, to skopírujeme na pásku $k-1$. Pásky 1 až $k-3$ vymažeme, vymažeme aj vstupnú pásku a dáme na ňu obsah pásky $k-2$ (teda pôvodný vstup). A rovnako ako v predchádzajúcom kroku spustíme stroj M_2, ktorý vstupnú pásku prerobí na $f_2(x)$.
3. Skopírujeme vstupnú pásku na pásku k a spustíme stroj, ktorý sčíta obsahy pások $k-1$ a k a výsledok napíše na vstupnú pásku (napríklad prerobením sčítania z predchádzajúceho riešenia). Výpočet úspešne ukončíme.

Vzorové riešenia úlohy 10.3

Najprv si rozoberme triviálny prípad, keď Σ je prázdna množina. Jediným možným vstupom je vtedy prázdna páska, ktorú TS bud vždy akceptuje alebo nikdy neakceptuje. Oba prípady sa zrejme dajú vyjadriť pomocou TS2 bez ďalších pracovných symbolov.

Ďalej predpokladajme, že Σ obsahuje aspoň jeden znak, nazvime ho A. Stroj TS_1 budeme simulovať pomocou TS_2 tak, že každý znak pracovnej abecedy Γ_1 nahradíme postupnosťou znakov A a $\sqcup$. Všetky postupnosti sú dĺžky n, pričom $2^n \geq |\Gamma_1|$. Kódovanie volíme tak, že znaku $\sqcup$ zodpovedá postupnosť znakov $\sqcup$, aby sme zachovali predpoklad, že nepoužitá cast pásky obsahuje iba znaky $\sqcup$. (TS_2 bude rozoznávať koniec vstupu nie podľa jedného znaku $\sqcup$, ale podľa postupnosti n znakov $\sqcup$ na správnej pozícii.)

Inštrukcie stroja TS_1 simulujeme tak, že TS_2 prečíta z pásky n znakov a podľa toho zistí, na ktorom simulovanom znaku sa nachádza. Na základe toho prepíše daných n znakov a posunie sa o n pozícií doľava alebo doprava. (Znak $\not{c}$treba spracovávať trochu upraveným spôsobom.)

Na tomto mieste končila väčšina správnych riešení. Pre úplnosť sa ešte hodno zamyslieť, či dokážeme pôvodný vstup na páske upraviť do vstupného formátu pre TS_2 bez použitia pomocných symbolov. Komplikácia spočíva v tom, že počas konverzie pásky jej zatiaľ nespracovaná cast nie je zarovnaná na násobky n polí pásky a teda potrebujeme nejako rozpoznávať koniec nespracovanej časti a začiatok spracovanej časti, bez použitia pomocných symbolov.

(Podobnú úvahu treba urobiť aj v riešeniach, kde navrhujete, aby TS_2 bol viacpáskový Turingov stroj. Áno, viacpáskový Turingov stroj dokážeme simulovať pomocou jednopáskového, ale z pohľadu tohto príkladu je dôležité, či to vieme urobiť aj bez použitia pomocných symbolov.)

Náznak riešenia: Za zatiaľ neskonvertovanou vstupnou časťou pásky ponecháme $2n$ medzier a za nimi bude nasledovat skonvertovaná cast pásky. Turingov stroj príde na koniec skonvertovanej časti pásky, posunie skonvertovanú cast o $n-1$ pozícií doprava, príde na koniec neskonvertovanej časti pásky a skonvertuje jeden znak. To opakuje, až spracuje celý vstup.

Využívame pritom poznatok, že v neskonvertovanej časti pásky sa žiadne medzery nenachádzajú a v skonvertovanej sa nikdy nenachádza $2n$ medzier za sebou. (Nestačí použiť iba n medzier, tie sa totiž môžu vyskytnúť aj náhodou na hranici dvoch znakov. Napríklad ak znak A nahradíme postupnosťou $A \sqcup \sqcup$ a znak B postupnosťou $\sqcup \sqcup A$, pri konverzii reťazca AB vznikne postupnosť $A \sqcup \sqcup \sqcup \sqcup A$, v ktorej sa vyskytujú viac ako tri medzery za sebou.)